```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node
{
   int data;
   struct Node *next;
} Node;

void insertAtBeginning(Node **head, int new_data);
void deleteAtBeginning(Node **head);
void deleteAtEnd(Node **head);
void delete(Node **prev_node, int pos);
void printList(Node *next);

void insertAtBeginning(Node **head, int new_data)
{
   Node *new_node = (struct Node *)malloc(sizeof(Node));
   new_node->data = new_data;
   new_node->next = *head;
   *head = new_node;
}

void deleteAtBeginning(Node **head)
{
   Node *ptr;
```

```
    if (head == NULL)
    {
        printf("\nList is empty");
    }
    else
    {
        ptr = *head;
        *head = ptr->next;
        free(ptr);
        printf("\n Node deleted from the beginning ...");
    }
}

void deleteAtEnd(Node **head)
{
    Node *ptr, *ptr1;

    if (*head == NULL)

    {

        printf("\nlist is empty");
    }

    else if ((*head)->next == NULL)
```

```
{

   free(*head);

   *head = NULL;

   printf("\nOnly node of the list deleted ...");
}

else

{

   ptr = *head;

   while (ptr->next != NULL)

   {

      ptr1 = ptr;

      ptr = ptr->next;
   }

   ptr1->next = NULL;
```

```
        free(ptr);

        printf("\n deleted Node from the last ...");
    }
}
void delete(Node **head, int pos)
{
    Node *temp = *head, *prev;

    if (temp == NULL)
    {
        printf("\nList is empty");
        return;
    }

    if (pos == 1)
    {
        *head = temp->next;
        free(temp);
        printf("\ndeleted node with position %d", pos);
        return;
    }

    for (int i = 0; temp != NULL && i < pos - 1; i++)
    {
        prev = temp;
```

```
        temp = temp->next;
    }

    if (temp == NULL)
    {
        printf("\nPosition out of range");
        return;
    }

    prev->next = temp->next;
    free(temp);
    printf("\ndeleted node with position %d", pos);
}
void PrintList(Node *node)
{
    while (node != NULL)
    {
        printf("%d\n", node->data);
        node = node->next;
    }
}

int main()
{
    int ch, new, pos;
    Node *head = NULL;
```

```
while (ch != 6)
{
   printf("Menu\n");
   printf("1.Insert into linked list\n");
   printf("2.Delete at beginning\n");
   printf("3.Delete at a specific position\n");
   printf("4.Delete at end\n");
   printf("5.Display linked list\n");
   printf("6.Exit\n");
   printf("Enter your choice\n");
   scanf("%d", &ch);
   switch (ch)
   {
   case 1:
   {
      printf("Enter the data you want to insert at beginning\n");
      scanf("%d", &new);
      insertAtBeginning(&head, new);
      break;
   }
   case 2:
   {
      deleteAtBeginning(&head);
      break;
   }
   case 3:
```

```
{
   printf("Enter the position at which you want to delete \n");
   scanf("%d", &pos);
   delete (&head, pos);
   break;
}
case 4:
{
   deleteAtEnd(&head);
   break;
}
case 5:
{
   printf("Created linked list is:\n");
   PrintList(head);
   break;
}
case 6:
{
   return 0;
   break;
}
default:
{
   printf("Invalid data!");
   break;
```

```
    }
    }
  }
  return 0;
}
```

```
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
1
Enter the data you want to insert at beginning
3
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
1
Enter the data you want to insert at beginning
2
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
1
Enter the data you want to insert at beginning
1
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
5
Created linked list is:
1
2
3
```

```
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
3
Enter the position at which you want to delete
1

deleted node with position 1Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
5
Created linked list is:
2
3
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
2

 Node deleted from the beginning ...Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
4

Only node of the list deleted ...Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
5
Created linked list is:
```

```
Menu
1.Insert into linked list
2.Delete at beginning
3.Delete at a specific position
4.Delete at end
5.Display linked list
6.Exit
Enter your choice
6
```